IJCSN International Journal of Computer Science and Network, May 2014
ISSN (Online) : 2277-5420
www.IJCSN.org

Conference Proceeding – RDFCORDGE - 2014

Neuro-Fuzzy Controlled Design and Optimization of Automatic Vehicle using VLSI

[1] K. L. Kar, [2] B.B. Mangaraj

[1] Asso. Prof., Department of Electronics and Telecommunication Engineering
MMCT, Raipur, C.G. India

[2] Asso. Prof., Department of Electronics Engineering
VSSUT, Burla, Odisha India

Abstract- In this paper I designed and optimize the performance neuro-fuzzy control Automatic Vehicle (AV) using VHDL. Hybrid neuro-fuzzzy technique is used for designing the system. Neural network (NN) is first designed using VHDL then the output of NN is feed into fuzzy logic controller [1]. There are different types of neuro-fuzzzy techniques. It is a vehicle which can guide itself from its current location to some destination without the aid of a driver or operator. Such vehicles are sometimes described as autonomous vehicles or mobile robots. These terms are often used interchangeably but in general an AV operates in a new environment or structured environment such as a warehouse or factory and is designed to carry out a specific task or tasks while mobile robots are often designed to exhibit intelligent behavior's to demonstrate the application of particular techniques[2].

This intelligent control system is to sense and perceive its surroundings necessary to support navigational task. There are steady progresses made from the earlier approach to current systems involving multiple sensors such as laser system, stereovision, GPS, etc. Recent advance in sensors, communication, and machine intelligence have made it possible to design more sophisticated Automated Guided Vehicle (AVG)[3]. The potential applications of AVG range from daily life, search and rescue mission to military during or after terror attack. In recent years, global warming effects have resulted in frequent disasters, such as: typhoon, hurricane, storms, etc. These disasters have posted serious challenges to the search and rescue teams during or after the disaster period.

Keywords- *VHDL, FPGA, AV, AVG, GPS,ANN, neuro-fuzzy, Hybrid Neuro-Fuzzy System, FR, CRCS, IR.*

1. Introduction

The human brain which consists of approximately 100 billion neurons that are connected by about 60 trillion connections forms the most complex structure known in the universe. Brain functions such as sensory information processing and cognition are the results of emergent computations carried out by these massive neural networks (NNs)[4]. Artificial NNs are computational models that are inspired by the principles of computations performed by the biological NNs of the brain [5]. The processing in the brain is mainly parallel and distributed: the information are stored in connections, mostly in myeline layers of axons of neurons, and, hence, distributed over the network and processed in a large number of neurons in parallel. The brain is adaptive from its birth to its complete death and learns from exemplars as they arise in the external world. NNs have the ability to learn the rules describing training data and, from previously learnt information, respond to novel patterns [6]. NNs are fault-tolerant, in the sense that the loss of a few neurons or connections does not significantly affect their behavior, as the information processing involves a large number of neurons and connections. Artificial neural networks (ANN) have found applications in many domains for example signal processing, image analysis, medical diagnosis systems [7], and financial forecasting [8].

Artificial neural systems can be considered as simplified mathematical models of brain like systems and they function as parallel distributed computing networks. However, in contrast to conventional computers, which are programmed to perform specific task, most NNs must be taught, or trained. They can learn new associations, new functional dependencies and new patterns [6]. The most important advantage of NNs is their adaptivity. NNs can automatically adjust their weights to optimize their behavior as pattern recognizers, decision makers, system controllers, predictors, etc. Adaptivity allows the NN to perform well even when the environment or the system being controlled varies over time. There are many control problems that can benefit from continual nonlinear modeling and adaptation.

The main advantages of NN over conventional systems are their ability to perform nonlinear input-output mapping, generalization, adaptivity and fault tolerance [7]. The development of fuzzy logic was motivated in large measure by the need for a conceptual framework which

can address the issue of uncertainty and lexical imprecision. In fuzzy logic, exact reasoning is viewed as a limiting case of approximate reasoning[5]. In this logic, everything is a matter of degree and knowledge is interpreted as a collection of elastic variables or equivalently, fuzzy constraint on a collection of variables. Inference is viewed as a process of propagation of elastic constraints. Any logical system can be fuzzified. There are two main characteristics of fuzzy systems that give them better performance for specific applications. First Fuzzy systems are suitable for uncertain or approximate reasoning, especially for the system with a mathematical model that is difficult to derive. Second Fuzzy systems allow decision making with estimated values under incomplete or uncertain information. While fuzzy logic performs an inference mechanism under cognitive uncertainty, computational NNs offer exciting advantages, such as learning, adaptation, fault tolerance, parallelism and generalization [6].

The main disadvantage of fuzzy system is that they do not have much learning capability to tune their fuzzy rules (FR) and membership functions. Normally FR are decided by experts or operators according to their knowledge or experiences. However when the fuzzy system model is designed, it is often too difficult (sometimes impossible) for human being to define all the desired fuzzy rules or membership functions in an optimized way, due to the ambiguity, uncertainty or complexity of the identifying system. Also, fuzzy systems do not have any learning capability in which their fuzzy rules, along with their corresponding membership function, could be automatically tuned in order to reach the desired optimal fuzzy rules and membership function.

To enable a system to deal with cognitive uncertainties in a manner more like humans, one may incorporate the concept of fuzzy logic into the NNs. The resulting hybrid system is called fuzzy neural, neural fuzzy, neuro-fuzzy or fuzzy-neuro network[8]. NNs are used to tune membership functions of fuzzy systems that are employed as decision-making systems for controlling equipment. Although fuzzy logic can encode expert knowledge directly using rules with linguistic labels, it usually takes a lot of time to design and tune the membership functions which quantitatively define these linguistic labels. NN learning techniques can automate this process and substantially reduce development time and cost while improving performance.

To overcome the problem of knowledge acquisition, NNs are extended to automatically extract fuzzy rules from numerical data. Cooperative approaches use NNs to optimize certain parameters of an ordinary fuzzy system, or to preprocess data and extract fuzzy control rules from data. The basic processing elements of NNs are called artificial neurons, or simply neurons. The signal flow into neuron inputs, x_i , is considered to be unidirectional as indicated by arrows as shown in figure 1, where the neuron's output signal is a function of input signals (x_i) and weights (w_i). All x's and w's are real numbers. The input neurons do not change the input signals but adjust the weight for desired result [4]. The signal x_i interacts with the weight w_i to produce the product $p_n = w_i.x_i$; i = 1,…….., n. The output information p_i's from first layer neurons are aggregated, by addition, to produce the input for neuron in the second layer. This second layer neuron will receive the net signal as shown in equation (1)

$$Net = p_1 + \ldots + p_n = w_1.x_1 + \ldots + w_n.x_n \ldots\ldots..(1)$$

The neuron uses its transfer function f, which could be a sigmoidal function as shown in equation (2),

$$f(t) = 1/(1 + e^{-t}) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

to compute the output (y) as follows

$$y = f(Net) = f(w_1.x_1 + \ldots\ldots\ldots + w_n.x_n) \ldots\ldots\ldots.. (3)$$

This simple NN, which employs multiplication, addition, and sigmoidal function f, will be called as regular (or standard) NN.

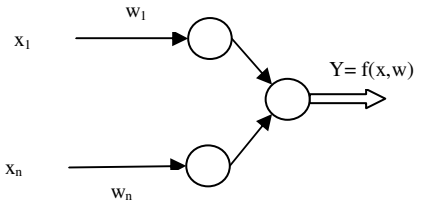

Figure 1: A simple neural net.

Automated Guided Vehicle (AGV) is an intelligent machine that has 'intelligence' to determine its motion status according to the local environmental restrictions. The successful operation of AGV needs to sense its local environment, be able to plan its maneuver and then act accordingly based on this plan. This type of system must be placed in a known space from which it can determine its orientation via the use of markers (e.g. white lines, humps, walls, obstacles, etc.). The model of the AGV itself may be easy to establish, however, the environmental model is difficult to obtain as it is dynamic in nature.
Currently three approaches for navigation of AGV are generally used: model-based approaches, sensor-based approaches, and hybrid approaches. Model-based approaches need an accurate description of the environment to generate an obstacle free path. Techniques for model-based path generation include road mapping cell decomposition, and potential fields. All of these methods

can generate a path from an initial point to the end point using a model of the environment. However, it is usually difficult to obtain an accurate model of a road with its Dynamic Local Environment (DLE). Sensor-based approaches execute control commands based on sensor data. A promising strategy for sensor-based approaches is the behavioral architecture, which consists of multiple behaviors, each one of which reacts to sensor input based on a particular concern of the navigation controller. Examples of behaviors include goal-attraction, wall or line-following, and obstacle-avoidance. The main advantage of sensor-based approaches is that the vehicle can travel safely in a changing environment, since it can react immediately to avoid obstacles detected by sensors. The major disadvantage of sensor-based approaches is that the vehicle may travel without a goal, even if the path to such a goal exists, such as when it loses its track because of DLE.

2. Hybrid Neuro-Fuzzy System

In this category, a neural network is used to learn some parameters of the fuzzy system (parameters of the fuzzy sets, fuzzy rules and the rules of weights) of a fuzzy system in an iterative way. The majority of the researchers use the neuro-fuzzy term to refer only hybrid neuro-fuzzy system[9].

2.1 Cooperative Neuro-Fuzzy Systems

In a cooperative system as shown in figure 2 the neural networks are only used in an initial phase. In this case, the neural networks determine sub-blocks of the fuzzy system using training data, after this, the neural networks are removed and only the fuzzy system is executed [10]. In these systems, the structure is in total not interpretable what can be considered a disadvantage.

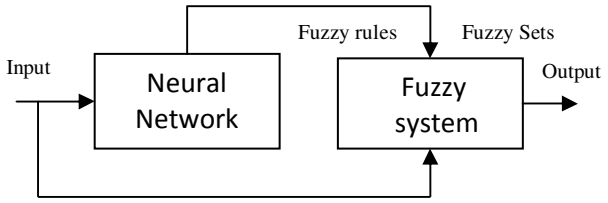

Figure 2: Cooperative Systems.

2.2 Concurrent Neuro-Fuzzy Systems

A concurrent system as shown in figure 3 is not a neuro-fuzzy system in the strict sense, because the neural network works together with the fuzzy system. This means that the inputs enters in the fuzzy system, are pre-processed and then the neural network processes the outputs of the concurrent system or in the reverse way. In the concurrent neuro-fuzzy systems, the results are not completely interpretable, what can be considered a disadvantage.

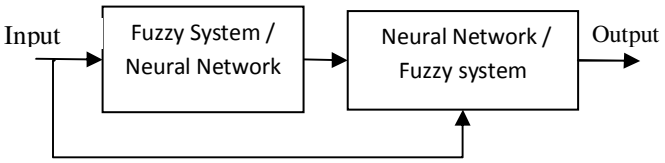

Figure 3: Concurrent Systems

2.3 Hybrid Neuro-Fuzzy Systems

A hybrid neuro-fuzzy system is a fuzzy system that uses a learning algorithm based on gradients or inspired by the neural networks theory (heuristical learning strategies) to determine its parameters (fuzzy sets and fuzzy rules) through the patterns processing (input and output)[11]. A neuro-fuzzy system can be interpreted as a set of fuzzy rules. This system can be entirely created from input output data or initialized with some priori knowledge in the same way of fuzzy rules. The resultant system by fusing fuzzy systems and neural networks has advantages of learning through patterns and the easy interpretation of its functionality. There are several different ways to develop hybrid neuro-fuzzy systems; therefore, being a recent research subject, each researcher has defined its own particular models. These models are similar in its essence, but they present basic differences.

Many types of neuro-fuzzy systems are represented by neural networks that implement logical functions. This is not necessary for the application of a learning algorithm into a fuzzy system; however, the representation through a neural network is more convenient because it allows us to visualize the flow of data through the system and the corresponding error signals that are used to update its parameters. The additional benefit is to allow the comparison of the different models and visualize its structural differences[12].

3. Design Methodology

3.1 Vehicle Guidance Systems

An Automated, Guided Vehicle (AGV) is a vehicle which can guide itself from its current location to some destination within its environment without the aid of a driver or operator. Such vehicles are sometimes described as mobile robots or autonomous vehicles[17]. These terms are often used interchangeably but in general an AGV operates in a structured environment such as a warehouse or factory and is designed to carry out a specific task or tasks while mobile robots are often designed to exhibit

'intelligent behaviors to demonstrate the application of particular techniques[2].

There are many elements which must come together before a vehicle can navigate successfully around its environment. These elements need to be able to cope with a changing environment and enable the vehicle to carry out its assigned task in an efficient manner. In order to navigate from a start position to a goal an AGV needs to have sensors to gather information on its environment and use this information either to produce an immediate reaction or in some form of path planning.

Typically sensor information is used to locate the vehicle and make a local map of its surroundings. The control system may also have access to a global map of its position. This information can be used to plan a path to the goal and information about the path can be passed to the navigation system. Sensor or mapping information may also be used by an obstacle avoidance system to influence the navigation controller. A low level motion control system may be used to translate navigation demands into actions. A schematic of some of the possible relationships between these elements is shown in Figure 4.

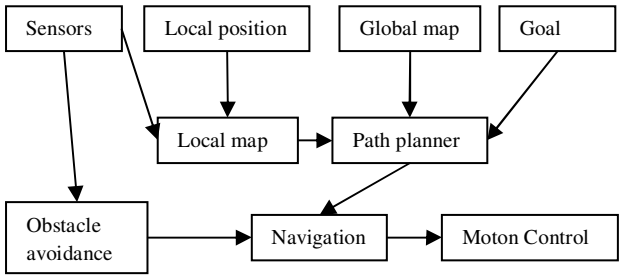

Figure 4: Schematic Diagram of the Elements of a Vehicle Guidance System.

AGV guidance systems may have some or all of these elements depending on their purpose and the approach favoured by their designers. The design of individual elements and how they can interact is discussed in more detail in the following sections which give a brief introduction to the some of the important elements of automated vehicle design.

The Compact Real time Control System (CRCS) of an ALV is shown in Figure 5. It consists of a remote control system, navigation system, vehicle control system, and multiple sensors that are connected to the system[13].

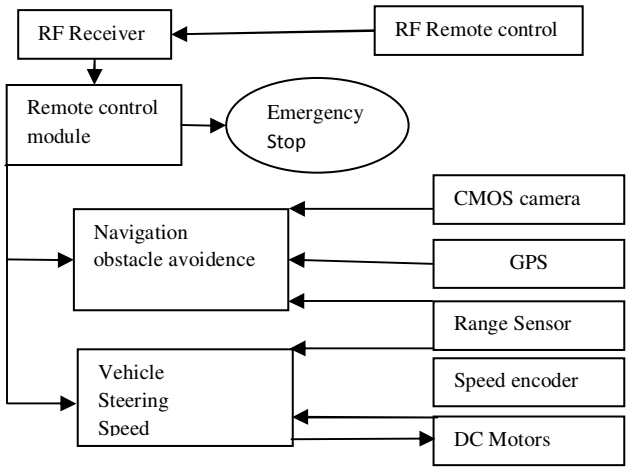

Figure 5: CRCS control architecture.

Navigation system is a high-level controller that performs obstacle avoidance. ALV can sense its surrounding with the range sensors and this information will be used for obstacle detection. Ultrasonic sensors are used in this work and it can be set to detect obstacles in a desired distance. Infrared (IR) sensor can also be used to sense range but it is easily influenced by ambient light at outdoor environment. Thus, ultrasonic sensor is preferred over IR sensor in the development of CRCS[14]. Besides, a color camera is used to acquire real-time image of the surrounding. The Global Positioning System (GPS) will provide the location of the ALV on the earth. The trajectory instructions are generated in this system and the control signals will be sent to the vehicle control system. In case the ALV faces an indecisive situation, the manual interaction will come into control where remote control system allows a shift from autonomous to manual operation dynamically.

The vehicle control system is a low-level controller that performs steering and driving tasks. It processes the signals from the navigation system, and the rotary encoders and then generates control signals to the DC motors. The rotary encoder will encode the speed of each DC motor and provide feedback to the system for speed control. One of the key features in this vehicle control system is zero-radius turning. The ALV is capable of a zero-radius turn in a confined area such as a tunnel.

3.2 Sensor Systems

Sensing is vital for the control of automated vehicles. Without sensor information it becomes impossible to find out where the vehicle is and plan a route to its goal. Sensors provide the information needed to locate the vehicle, map its surroundings and avoid collisions. There is a wide variety of sensors used in robotics the most

common used sensors are ultrasonic transducers, sonar arrays, infra red sensors, lasers, vision, odometry etc.

4. Front End Design Methodology

Hardware description language (HDL) is a language for formal description and design of electronic circuits, and most commonly, digital logic circuits. It can describe the circuit's functionality, its design and organization, and tests can be created to verify its operation by means of simulation. In this work, the use of VHDL and Verilog for logic synthesis has several advantages over other design implementation methods because designs can be rapidly prototyped as the FPGA device is an easily reconfigurable logic device[2][15].

The constructs of the VHDL code for synthesis can have a great effect on the system's performance. Intellectual Pro perty (IP) building blocks in the design software have been rigorously tested and meet the exacting requirements of various industry standards. Therefore, the use of IP blocks in the system can guarantee the system performance. However, for some non-standard piece of code constructs, it might cause the synthesis tool to try a non-optimal implementation algorithm, producing synthesized logic of lower quality. Furthermore, targeting the predefined logic structures of FPGA which has fixed properties requires a special design approach when writing the VHDL or Verilog code[16].

4.1 Simulation

In typical design flow, simulations must be run on the individual functional block to check on the block's functionalities before system integration. After the system integration to create the CRCS, simulations are then performed on the complete system to observe the system behaviors. If a faulty custom functional block is not verified and it is integrated into CRCS, it is difficult to detect the fault from multiple blocks in the later design stage, and system performance is not guaranteed. So, simulations are useful to meet certain needs, including the following cases:

To verify the functionalities of a custom component before implementing it on hardware.
To verify the cycle-accurate performance of a system before target hardware is available.

4.2 Vector Waveform File

Vector Waveform File (VWF) describes the simulation input vectors and simulation output vectors as graphical waveforms. Waveform Editor is used to view and edit VWF. During simulation, the VWF is an input file only. The Simulator requires a VWF to provide the input vectors that drive simulation. In order for a VWF to be used in creating stimulus for the Simulator, it must specify the following:

The input logic levels (vectors) that drive the input pins and determine the internal logic levels throughout the design. The nodes to be observed start and stop times for applying vectors, intervals at which vectors are applied. The radix used to interpret logic levels.

4.3 Custom Component Block Level Simulation

There are multiple blocks used to build the complete system. As a demonstration, simulation that was performed on the FSM block in vehicle control module is shown in Figure 6.The FSM is a control mechanism that ensures safety operations of the DC motors so it must be verified before implementation. Table 1 lists the binary values of each FSM state in the system.

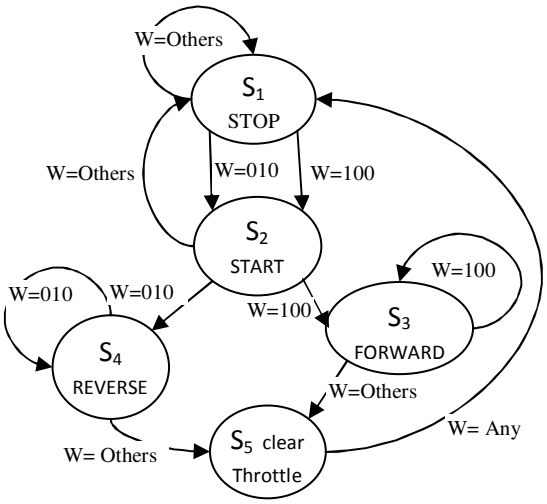

Figure 6: FSM in vehicle control module.

Table 1: Binary value of FSM states.

State	State value (binar
Stop/Brake	000
Clear Brake	001
Forward	010
Reverse	011
Clear Throttle	100

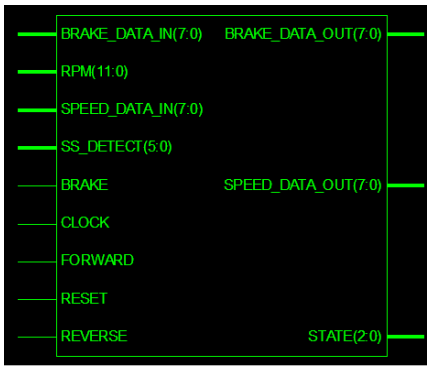

Figure 7: RTL of Vehicle control module FSM.

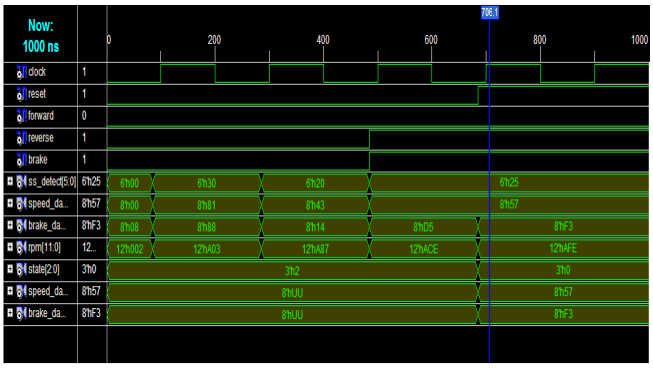

Figure 8: Vehicle control module FSM simulation.

Figure 7 shows of Vehicle control module FSM. Figure 8 shows the simulation of the FSM in vehicle control module. The input vectors for this simulation are CLOCK, RESET, FORWARD, REVERSE, BRAKE, RPM. All these input vectors were defined and set in Waveform Editor of Xilinx 9.1i before simulation. For the first rising clock edge, there are no activities in the input signals thus FSM remains in Stop/Brake (000) state. At the second rising clock edge, the control system asserts FORWARD to logic „high" and FSM jumps to Clear Brake (001) state to prepare the vehicle for acceleration. Finally in the third rising clock edge, FORWARD signal is still logic „high" thus FSM makes a transition to Forward (010) state and vehicle accelerates to forward direction.

This simulation shows that FSM behaviors meet design expectations. It makes a smooth and correct transition between states. In case error is found from simulation, input vectors are checked to detect any incorrect input that causes unpredictable behavior. If error is not arising from incorrect input vectors, component coding needs to be evaluated. Since this simulation is focused on the functional verification instead of timing behavior, so the timescale is set to unit of micro-second in order to reduce the simulation time. A larger time interval and time unit will increase the simulation time significantly. However, if the design requires accurate timing behaviors from simulation, the actual time scale has to be set.

4.4 System Level Simulation

After the simulations on each component block, all blocks are connected using on-chip system interconnects to establish communications between blocks. The complete system is verified by simulation. Figure 9 shows simulation which consists of inputs to the vehicle control system and the outputs after processing through internal logics of multiple blocks.

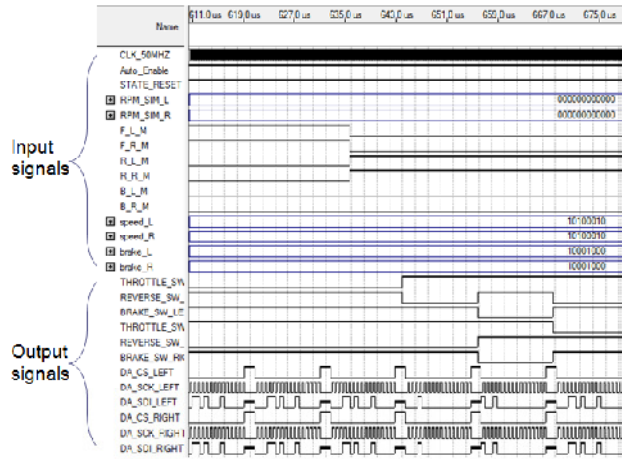

Figure 9: CRCS system level simulation.

Before the simulation is run, input vectors are designed and defined so that it can test various functionalities of the system[16]. At this stage, system outputs are known based on the design specifications without running the simulation. The expected outputs are used to verify the simulation result, as indicated by the output signals in Figure 8. The outputs that are of utmost importance in the system are control signals and data to configure the external Digital-to-Analog Converter (DAC) chip which controls DC motors. These signals are labeled with "DA" in front of the signal name in the simulation result. All waveforms are studied vehicle fully to ensure that it meets design specifications. From simulation, it shows that system sends correct configuration data to DAC chip through SPI communication.

5. Conclusion

A neuro-fuzzy control Automatic Vehicle (AV) navigation system has been developed which can be fed with information gathered from the environment using an array of sensors. Ultrasonic sensors are to provide the distance of an obstacle from the AGV whereas a GPS device and heading compass are to compute the orientation of the AGV with respect to the target destination (heading

angle). The output of the proposed navigation system is the steering angle and velocity of the AGV. These outputs are then used as stimulus for producing the required mechanical behavior through the system hardware. The hardware consisting of a mounting frame, driving and driven gears and servo motors is required. Firstly, the neural network approach can be further optimized to reduce the mean square error in the training. There exist a vast number of algorithms for machine learning which can be explored. Additionally, the ultrasonic sensor data collection system can be replaced by a visual sensors or stereoscopic cameras. The data collected from cameras would also require similar processing. Image processing techniques and algorithms through artificial neural networks, neuro-fuzzy systems and genetic algorithms are an area of high interest. Developments in these fields can be incorporated with those in the autonomous decision making systems for further improvements in AVG.

References

[1] Svehicleselli Franco, Tsoi Ah Chung; "Universal Approximation Using Feedforward Neural Networks: A Survey of Some Existing Methods, and Some New Results"; Neural Networks, Vol. 11, No. 1, pp. 15–37, 1998.

[2] Harisha S.K., Kumar Ramkanth P., Sharma S.C.; "Fuzzy Logic Reasoning To Control Mobile Robot On Pre- Defined Strip Pat; Proceedings of World Academy of Science, Engineering and Technology, Vol. 4, pp.539- 546,2003.

[3] D. Nauck, R, Kurse, "Neuro-FuzzySystems for Function Approximation", 4th International Workshop Fuzzy-Neuro Systems, 1997.

[4] K.L. Kar at. el., "VHDL implementation of an ANN" published in ELSEVIER , Volume 4, Issue 5, May 2012, ISBN: 978-93-5107-194-5 531-537.

[5] L.A. Zadeh, Fuzzy Sets, Information and Control, 1965.

[6] L.A. Zadeh, Outline of A New Approach to the Analysis of of Complex Systems and Decision Processes, 1973.

[7] L.A. Zadeh, "Fuzzy algorithms," Info. & Ctl., Vol. 12, 1968, pp. 94-102.

[8] D. Nauck, "Beyond Neuro-Fuzzy Systems: Perspectives and Directions". Proc. of the Third European Congress on Intelligent Techniques and Soft Computing (EUFIT'95), Aachen, 1995.

[9] D. Nauck; "A Fuzzy Perceptron as a Generic Model for Neuro-Fuzzy Approaches". Proc. Fuzzy-Systems, 2nd GI- Workshop,I Munich, 1994.

[10] S. Sulzberger, N. Tschichold e S. Vestli, "FUN: Optimization of Fuzzy Rule Based Systems Using Neural Networks", Proceedings of IEEE Conference on Neural Networks, San Francisco, March 1993, pp. 312-316.

[11] D. Nauck, F. Klawon; R. Kruse, "Foundations of Neuro-Fuzzy Systems", J.Wiley & Sons, 1997.

[12] Svehicleselli Franco, Tsoi Ah Chung; "Universal Approximation Using Feedforward Neural Networks: A Survey of Some Existing Methods, and Some New Results"; Neural Networks, Vol. 11, No. 1, pp. 15–37, 1998.

[13] Davis Ian Lane; "A Modular Neural Network Approach to Autonomous Navigation"; The Robotics Institute, Vehiclenegie Mellon University, Pittsburgh, PA; April 1996.

[14] Liu K. & Lewis F.L.; "Fuzzy Logic based Navigation Controller For An Autonomous Mobile Robot; IEEE, Vol. 7, pp.49-63,1989.

[15] Parasovic A., latinovic I . , "A Neural Network FPGA Implementation" . IEEE. 5th seminar on Neural Network Application in Electrical Engineering , September 2000 , PP117 – 120.

[16] Cirstea, M.N., Dinu, A., Nicula, D.: A Practical Guide to VHDL Design, Editura Tehnica, Bucharest, Romania, 2001, ISBN: 973–31-1539–8.

[17] Castillo Osvehicle, Aguilar Luis T., Vehicledinas Selene; "Fuzzy Logic Tracking Control For Unicycle Mobile Robots"; Engineering Letters, pp. 73-82, 2003.

Authors Profile

Mr. K. L. Kar born in West Bengal, India. He received his B. Tech. degree in electronics from N.I.T., Rourkela, India in 2003, M.E.(VLSI Design), from C.S.V.T.U., India in 2009. He joined as an Asso. Prof. in E&TC department, MMCT, Raipur and working as a research engineer in A.M.G., Toronto, Canada since 2008. Currently doing PhD from Jadavpur University, Kolkata, India. He is a member of ACEEE, IDES. His present interests are VLSI Design, Neural Network, Fuzzy Logic and Programming with VHDL and Verilog.

Dr. B. B. Mangaraj received the BE degree from University College of Engineering, Burla, Orissa, India, in 1994 and the ME TeLE degree and Ph. D. (Engg.) degree from Jadavpur University, Kolkata, India, in 2003 and 2012 respectively. In 2006, he joined the Department of Electronics and Telecommunication Engineering, University College of Engineering, Burla, as a Lecturer. In 2011 he joined as a Reader again in the same department. His present interests are in Analysis, design, coding and optimization of wire antenna structures and micro-strip antennas starting from simple to complicated one. He is also taking interest to apply VLSI in RF Engineering. He has been honored with university medal for his ME course by Jadavpur University.